

**SPRAWOZDANIE**  
**Organizacja i Architektura Komputerów**  
**Projekt**

### **I. Cele projektu**

Napisać program realizujący zadanie interpretera języka BF.

### **II. Założenia**

1. Program został napisany w całości w języku assembly.
2. W programie znajduje się opcja wczytywania pliku źródłowego z kodem BF.
3. W programie znajduje się opcja wczytywania kodu BF z konsoli.
4. Program zawiera w sobie menu umożliwiające wybranie źródła kodu przeznaczonego do interpretacji.

### **III. Język BF**

#### **1. Geneza**

Język BF (skrót od Brainf\*ck) jest językiem ezoterycznym programowania, który został stworzony około 1993 roku przez Urbana Müllera. Składnia języka jest bardzo uboga co wyróżnia go na tle innych. Interpretacja opiera się na prostym modelu maszyny taśmowej z przesuwным wskaźnikiem (idea podobna do maszyn Turinga).

#### **2. Instrukcje języka BF ze swoimi odpowiednikami w języku C**

Znak	Znaczenie	Odpowiednik w C
>	inkrementuje wskaźnik	++p
<	dekrementuje wskaźnik	--p
+	inkrementuje wartość w komórce wskazywanej przez wskaźnik	++(*p)
-	dekrementuje wartość w komórce wskazywanej przez wskaźnik	--(*p)
.	wyświetla znak w ASCII o wartości wskazywanej komórki	putchar(*p)
,	pobiera znak i wstawia go do wskazywanej komórki	*p=getchar()
[	początek pętli, która nie wykona się jeżeli w aktualnie wskazanej komórce znajduje się wartość 0	while(*p){
]	koniec pętli	}



Kluczowy fragment kodu odpowiedzialnego za wczytywanie danych z pliku:

```
movq 8(%rsi), %rdi # path
movl $0, %esi      # flags
movl $0666, %edx   # mode
movl $2, %eax      # SYS_OPEN
syscall
movl %eax, %ebx    # save fd

# read the file
movl %eax, %edi    # fd
leaq buf, %rsi     # buf
movl $bufSize, %edx # count
movl $0, %eax      # SYS_READ
syscall

# close the file
movl %ebx, %esi    # fd
movl $3, %eax      # SYS_CLOSE
syscall

exit:
xor %eax, %eax     # return 0
pop %rbx           # restore rbx
```

### Etap 3.

Dodać możliwość wczytywania kodu programu BF z konsoli, oraz proste menu umożliwiające wybranie źródła do interpretacji (z pliku lub z konsoli). Do wczytywania z konsoli ciągu znaków przeznaczonego do interpretacji została wykorzystana funkcja systemowa SYS\_READ.

Kluczowy fragment kodu odpowiedzialnego za wczytywanie ciągu znaków z konsoli:

```
#pobranie tekstu do dalszej
#obróbki
movq $SYSREAD, %rax
movq $STDIN, %rdi
movq $buf, %rsi
movq $bufSize, %rdx
syscall
```

## **V. Wnioski**

Realizacja projektu pozwoliła na lepsze zrozumienie języka assembly oraz tego jak działa komputer od środka. Ponad to umożliwiła poszerzenie wiedzy o pracy samego systemu operacyjnego.